

//Guillermo Escobar P 256243

2. Which of the following decimal numbers has an exact representation in binary notation?

- (A) 0.1                      (B) 0.2                      (C) 0.3                      (D) 0.4                      (E) 0.5

Puesto que la representación binaria se conoce en dos en dos formas, primero la exacta en una combinación de 0 y 1 finita, y la otra como un numero representado también como 0 y 1 pero en una representación periódica infinita, conociendo esto el único numero de las opciones que se puede representar de ambas formas, exacta e inexacta es 0,5

Rta/ E

3. Bob writes down a number between 1 and 1,000. Mary must identify that number by asking "yes/no" questions of Bob. Mary knows that Bob always tells the truth. If Mary uses an optimal strategy, then she will determine the answer at the end of exactly how many questions in the worst case?

- (A) 1,000                      (B) 999                      (C) 500                      (D) 32                      (E) 10

Dado que se usa el algoritmo de búsqueda binario, disminuyendo así las posibilidades a la mitad en cada pregunta, en este caso para un arreglo de longitud 1000 el numero de preguntas en el peor caso es 10 .

Rta/ E.

7. Consider the following Pascal-like program fragment.

```
var i, j : integer ;
procedure P(k, m : integer) ;
begin
  k := k - m ;
  m := k + m ;
  k := m - k
end ;
i := 2 ;
j := 3 ;
P(i, j) ;
```

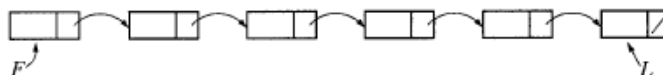
If both parameters to  $P$  are passed by reference, what are the values of  $i$  and  $j$  at the end of the program fragment?

- (A)  $i = 0, j = 2$                       (B)  $i = 1, j = 5$                       (C)  $i = 2, j = 3$                       (D)  $i = 3, j = 2$   
(E) None of the above

Puesto que ambos valores se ingresan por referencia el algoritmo modifica sus valores originales lo cual da como resultado después de ejecutarse los valores  $i = 3, j = 2$

Rta/ D

9. Consider a singly linked list of the form



where  $F$  is a pointer to the first element in the list and  $L$  is a pointer to the last element in the list. The time of which of the following operations depends on the length of the list?

- (A) Delete the last element of the list.  
(B) Delete the first element of the list.  
(C) Add an element after the last element of the list.  
(D) Add an element before the first element of the list.  
(E) Interchange the first two elements of the list.

Se trabaja con una lista enlazada simple y allí se tiene un apuntador tanto al primero como al ultimo elemento debemos suponer que al adicionar o eliminar elementos al principio o al final que es donde se encuentran los apuntadores no hará diferencia la longitud de la lista, sin embargo al intercambiar los dos primero elementos si la lista es de longitud 2 el intercambio se haría en un solo paso

Rta/ E

```
10.  p := 1; k := 0;
      while k < n do
      begin
        p := 2 * p;
        k := k + 1
      end;
```

For the program fragment above involving integers  $p$ ,  $k$ , and  $n$ , which of the following is a loop invariant; i.e., true at the beginning of each execution of the loop and at the completion of the loop?

- (A)  $p = k + 1$       (B)  $p = (k + 1)^2$       (C)  $p = (k + 1)2^k$       (D)  $p = 2^k$       (E)  $p = 2^{k+1}$

Para el segmento de código p calcula una potencia de 2 y k cuenta el numero de veces que se ha ejecutado el ciclo por lo tanto el invariante es  $p = 2^k$

Rta/ D

Questions 13-14 relate to the following C-like program.

```
#include <stdio.h>
main()
{
  float sum = 0.0, j = 1.0, i = 2.0;
  while (i/j > 0.001)
  {
    j = j + j;
    sum = sum + i/j;
    printf("%f\n", sum);
  }
}
```

13. How many lines of output does the program produce?

- (A) 0-9      (B) 10-19      (C) 20-29      (D) 30-39      (E) More than 39

14. Which of the following is the integer that best approximates the last number printed?

- (A) 0      (B) 1      (C) 2      (D) 3      (E) 4

Si se hace una prueba de escritorio al algoritmo vemos que el ciclo se ejecuta 8 veces o se a en la pregunta 13 nos da A)0-10 y en la ultima iteración da aproximadamente 2.97 lo que esta mas cerca en la pregunta 14 a 3

Rrat/ D

15. An integer  $c$  is a common divisor of two integers  $x$  and  $y$  if and only if  $c$  is a divisor of  $x$  and  $c$  is a divisor of  $y$ . Which of the following sets of integers could possibly be the set of all common divisors of two integers?

- (A)  $\{-6, -2, -1, 1, 2, 6\}$   
 (B)  $\{-6, -2, -1, 0, 1, 2, 6\}$   
 (C)  $\{-6, -3, -2, -1, 1, 2, 3, 6\}$   
 (D)  $\{-6, -3, -2, -1, 0, 1, 2, 3, 6\}$   
 (E)  $\{-6, -4, -3, -2, -1, 1, 2, 3, 4, 6\}$

Al revisar las posibilidades de respuesta podemos descartar A) por que todo numero divisible por 6 es divisible por 3 el cual no esta, B) y D) porque ningún numero es divisible por 0, por ultimo E) no cumple la condición de ser el conjunto de todos los divisores pues la lista son divisores de 12 pero faltaría el 12, lo cual nos deja que la correcta es C pues es el conjunto de todos los divisores enteros de 6.

Rta/ C

17. A particular parallel program computation requires 100 seconds when executed on a single processor. If 40 percent of this computation is "inherently sequential" (i.e., will not benefit from additional processors), then the theoretically best possible elapsed times for this program running with 2 and 4 processors, respectively, are
- (A) 20 and 10 seconds
  - (B) 30 and 15 seconds
  - (C) 50 and 25 seconds
  - (D) 70 and 55 seconds
  - (E) 80 and 70 seconds

Como el 40% del cómputo es totalmente secuencial lo cual significa que es ejecutado por un solo procesador inferimos que al aumentar la cantidad de procesadores para ejecutar el mismo procedimiento solo tendrá efecto sobre el 60% restante, por lo tanto la relación que describe el comportamiento para 2 y 4 procesadores respectivamente es:

$$40 + 60/2 = 70$$

$$40 + 60/4 = 55$$

Por lo tanto la respuesta correcta es 70 y 55 segundos

Rra/ D.

20. Let  $P$  be a procedure that for some inputs calls itself (i.e., is recursive). If  $P$  is guaranteed to terminate, which of the following statements must be true?
- I.  $P$  has a local variable.
  - II.  $P$  has an execution path where it does not call itself.
  - III.  $P$  either refers to a global variable or has at least one parameter.
- (A) I only
  - (B) II only
  - (C) I and II only
  - (D) II and III only
  - (E) I, II, and III

Y que la condición es que el procedimiento termine deben cumplirse alguna de las tres condiciones pero no necesariamente varias, podría cumplirse solo una, por esto la respuesta es I, II y III

Rta/ E.

26. Let  $S$  be the statement:

```
for  $i := 1$  to  $N$  do  $V[i] := V[i] + 1$ 
```

Which of the following perform(s) the same changes to  $V$  as  $S$ ?

- I.  $i := 0$  ;  
**while**  $i \leq N$  **do**  
**begin**  $i := i + 1$  ;  $V[i] := V[i] + 1$  **end**
- II.  $i := 1$  ;  
**while**  $i < N$  **do**  
**begin**  $V[i] := V[i] + 1$  ;  $i := i + 1$  **end**
- III.  $i := 0$  ;  
**while**  $i < N$  **do**  
**begin**  $V[i + 1] := V[i + 1] + 1$  ;  $i := i + 1$  **end**

- (A) I only
- (B) II only
- (C) III only
- (D) II and III only
- (E) I, II, and III

Si considero que  $S$  se ejecuta como for ( $i=1$  ;  $i \leq N$  ;  $i++$ ) y no como for ( $i=0$  ;  $i < N$  ;  $i++$ ), nótese la diferencia de una iteración, la única respuesta que coincidiría en el número de iteraciones sería la III, por esto la respuesta correcta es III only  
 Rta / C.

```
27. var  $i, j, x$  : integer ;  

read( $x$ ) ;  

 $i := 1$  ;  $j := 1$  ;  

while  $i < 10$  do  

begin  

 $j := j * i$  ;  

 $i := i + 1$  ;  

if  $i = x$  then exit  

end
```

For the program fragment above, which of the following statements about the variables  $i$  and  $j$  must be true after execution of the fragment?

- (A)  $(j = (x - 1)!) \wedge (i \geq x)$
- (B)  $(j = 9!) \wedge (i = 10)$
- (C)  $(j = 10!) \wedge (i = 10)$
- (D)  $((j = 10!) \wedge (i = 10)) \vee ((j = (x - 1)!) \wedge (i = x))$
- (E)  $((j = 9!) \wedge (i \geq 10)) \vee ((j = (x - 1)!) \wedge (i = x))$

Evaluando algoritmo para  $x \geq 10$ :

```

{i = 1, j = 1} j = j * i {i = 1, j = 1}
{i = 1, j = 1} i = i + 1 {i = 2, j = 1}
{i = 2, j = 1} j = j * i {i = 2, j = 2}
{i = 2, j = 2} i = i + 1 {i = 3, j = 2}
{i = 3, j = 2} j = j * i {i = 3, j = 6}
{i = 3, j = 6} i = i + 1 {i = 4, j = 6}
{i = 4, j = 6} j = j * i {i = 4, j = 24}
{i = 4, j = 24} i = i + 1 {i = 5, j = 24}
{i = 5, j = 24} j = j * i {i = 5, j = 120}
{i = 5, j = 120} i = i + 1 {i = 6, j = 120}
{i = 6, j = 120} j = j * i {i = 6, j = 720}
{i = 6, j = 720} i = i + 1 {i = 7, j = 720}
{i = 7, j = 720} j = j * i {i = 7, j = 5040}
{i = 7, j = 5040} i = i + 1 {i = 8, j = 5040}
{i = 8, j = 5040} j = j * i {i = 8, j = 40320}
{i = 8, j = 40320} i = i + 1 {i = 9, j = 40320}
{i = 9, j = 40320} j = j * i {i = 9, j = 362880}
{i = 9, j = 362880} i = i + 1 {i = 10, j = 362880}

```

Con esto resolvemos que después de la ejecución el algoritmo cumple:

$$(j = 9! \wedge i = 10) \vee (j = (x - 1)! \wedge i = x)$$

Ninguna de las respuestas comprende la respuesta exacta entonces seleccionamos la que mas se parece .

Rra/ B